REMARKS

This amendment is responsive to the Final Office Action[1] (hereinafter "Office Action") mailed on November 18, 2004. Claims 1-89 were presented for examination and were rejected. Claims 1, 4, 12, 15, 46, 49, 52, 55, 59, 63, 64, 68, 69, 70, 71, 75, 76, 82 and 83 are amended. No claims are added. No claims are deleted. Thus, claims 1-89 are pending. Claims 1, 12, 24, 37, 49, 52, 59, 63, 68, 70, 75 and 81 are independent claims.

Preliminarily, in the Office Action, paragraph 5, where a listing of claims that were amended in the prior response filed August 2, 2004 is presented, that listing is not complete. For the record, claims 11, 23, 51, 74 and 88 were also amended in that prior response.

35 U.S.C. § 112 REJECTION:

The Office Action suggests that each of claims 1, 4, 12, 15, 49, 52, 59, 63, 68, 70, and 75 do not have sufficient antecedent basis. The Office Action provides examples of the alleged insufficient antecedent basis and all examples appear to suggest that the alleged insufficiency is with respect to not modifying "architecture" with the term "software". Although Applicants respectfully disagree that the antecedent basis was insufficient since the claims were not ambiguous as they were presented, Applicants have nevertheless included "software" as a modifier of the term "architecture" in each of the

[1] The Office Action may contain a number of statements characterizing the cited references and/or the claims which Applicants may not expressly identify herein. Regardless of whether or not any such statement is identified herein, Applicants do not automatically subscribe to, or acquiesce in, any such statement.

above-enumerated claims, as well as in any other pending claim which recited the term "architecture" but did not have "software" as a modifier. Accordingly, any antecedent basis issue has been addressed and it is respectfully requested that this rejection be withdrawn.

35 U.S.C § 103(a) REJECTION:

Claims 1-62 and 75-89 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,086,622 to Abe et al., (hereinafter "Abe"), in view of "admitted prior art"[2] and further in view of U.S. Patent No. 5,295,256 to Bapat (hereinafter "Bapat"). Claims 63-74 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Abe in view of "admitted prior art". Applicants respectfully traverse these rejections for the following reasons.

Introduction

It is Applicants' position that the Examiner should not be applying Abe against Applicants' claims because Abe does not disclose or suggest at least: (1) conversion between two different *software* architectures as claimed (Abe discloses conversion between two different *hardware* architectures) and (2) management software, and certainly not management software as recited in Applicants' claims. Further, the "admitted prior art" and Bapat taken either alone, or in combination, do not cure these deficiencies in Abe, and each reference is not properly combinable with Abe in any event

[2] Applicants take issue with this phraseology. As expressed in the prior response filed on August 2, 2004, page 29, "Accordingly, although Applicants adopt herein the Examiner's phraseology 'admitted prior art' in reference to both Figs. 2A/2B and schema included therein for purposes of facilitating communication on the record, Applicants do not necessarily agree that Figs. 2A/B are prior art vis-à-vis the claimed subject matter."

for reasons given below.

A principal purpose of Applicants' invention is to provide a translator-compiler for converting legacy management software compatible with legacy or proprietary *software* architecture (software architecture such as that shown in Applicants' Fig. 2A) to be compatible with preferred, standard, non-legacy *software* architecture (software architecture such as that shown in Applicants' Fig. 2B). Abe has absolutely nothing to do with this conversion. Abe, by contrast, teaches compiling a high level language operable with a first *hardware* architecture into machine code operable with a second and non-existent *hardware* architecture, along with some other steps of decompiling and linking for purposes of allowing debugging of the machine code prior to availability of the second hardware architecture for which it is actually intended. Thus, Applicants' purpose relates primarily to conversion between two software architectures, but Abe's purpose relates primarily to debugging machine code. These goals are completely unrelated to each other. The detail of these differences shall be developed in the following discussion.

All Independent Claims: 1, 12, 24, 37, 49, 52, 59, 63, 68, 70, 75 and 81
Software Architecture Issue:

Preliminarily, in the Office Action, the "Examiner's response" on page 21 states: "The limitation regarding the software architecture in the preamble is not given any patentable weight because the body of the claim does not recited any limitation related to software architecture." Applicants respectfully disagree that the preamble should be ignored when considering a claim's limitations during examination. Nevertheless, for purposes of advancing the prosecution of this application, Applicants have amended claims that recite the term "architecture" by modifying that term with "software" to

achieve "software architecture" in every instance. Accordingly, Applicants respectfully request that the Examiner now give patentable weight to this limitation, in accordance with the Examiner's above-quoted statement in the Office Action.

Abe does not disclose or suggest first and second software architectures. As noted, Abe is hardware-oriented and is directed to the debugging of machine code to be used on new hardware architecture prior to availability of the new hardware architecture.

> "When a *computer of a new architecture* which is different from a present architecture is developed, it is necessary to debug a *machine program* for the computer. However, in general, when the machine program for the new computer is to be debugged, *a new computer (hardware)* is not completely produced, i.e., does not actually exist. Therefore, it is impossible to actually debug the machine program on the new computer." (Abe, column 1, lines 17-24; Emphasis added)

Thus, the problem to which Abe is directed has to do with hardware that will not be ready until a future date. Abe wants to be able to debug a *machine program* intended for the non-existing, new hardware architecture on presently-existing, similar hardware architecture. A machine program is machine code or "1's" and "0's" and is the code which the hardware understands and which makes it operate.

> "In the following descriptions, a computer of an existing architecture (hereinafter referred to as a first architecture) is a *serial executing type* and a computer of a new architecture (hereinafter referred to as a second architecture) is a *parallel executing type*." (Abe, column 4, lines 34-39, Emphasis added)

Thus, the example relied upon throughout Abe is converting from a *serial hardware architecture* to a *parallel hardware architecture*, where the serial machine *is* in existence but the new, parallel machine *is not*. Abe has a high level language program which it compiles for the new, parallel architecture; Abe decompiles it and reassembles it to run

on the currently-available, serial architecture to test the new machine code on the existing serial architecture, rather than wait for the parallel architecture to be ready.

Thus, in view of the above, when Abe refers to converting a program for a computer of a first architecture to a program for a computer of a second architecture, he is referring to a first *hardware* architecture and a second *hardware* architecture. As further evidence of the limitation of the disclosure in Abe to hardware architecture, consider:

> "a first step of *compiling* a first high-level language source program for a computer of a first architecture, *thereby producing a machine program for a computer of a second architecture*" (Abe, column 2, lines 5-8, and independent claims 1 and 10, emphasis added).

In other words, Abe discloses and recites in every claim, a step whereby he compiles a high level language (converts it into machine code). This high level language is for a computer having a *first* architecture. This compiling step produces a machine program or machine code. But it produces machine code for a computer having a *second* (different) architecture. The only kind of architecture that could possibly fit this description is *hardware* architecture because one does not and cannot produce machine code ("1's" and "0's") from software architecture for software architecture. Software architecture doesn't run machine code; only hardware architecture runs machine code. Therefore one could not produce machine code for a second architecture based solely on compiling a high level language for a first architecture *unless* the second architecture is *also* hardware architecture. It is thus clear that hardware architecture exclusively, and not software architecture, is being referred-to in Abe.

By contrast, Applicants' disclosure and claims are limited to *software*. The title is: "Translator-Compiler for Converting Legacy Management *Software*" (Emphasis

added). Software is discussed throughout the application: In reference to "legacy or proprietary architecture" Applicants' disclosure says: "These architectures are combinations of *software*, such as schemas, languages, and protocols, etc." (application, page 4, lines 2-3). In Figs. 2A and 2B, legacy *architecture* and CIM/XML *architecture* are depicted and these are the kinds of architectures to which Applicants' claims relate. Clearly, these are architecture stacks of software; no hardware is depicted in Figs. 2A/2B.

Consider Applicants' amended claims. Each and every independent claim has been amended to recite "software architecture" not only in each preamble (previously amended), but now also in the body of each claim in response to the Examiner's guidance. For example, claim 1 recites, *inter-alia*: "A computer system employing management software written in a first computer language compatible with first *software* architecture and not compatible with second *software* architecture, said system comprising: a schema formed within said first *software* architecture.....and, means for converting said called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said second *software* architecture." This software architecture limitation clearly defines around the disclosure of Abe, the primary or teaching reference, which is limited to hardware architecture as demonstrated above. Both "admitted prior art" and Bapat do not, and cannot, cure this hardware architecture deficiency of Abe; neither "reference" in combination with Abe provides a teaching or suggestion of Applicants' claimed subject matter.

Accordingly, all independent claims are not disclosed or suggested by Abe in combination with either or both "references" taken alone or together and it is respectfully

requested that the 35 U.S.C. § 103(a) rejection of these claims be withdrawn and the claims allowed.

Nevertheless, consider the requirements for a prima facie case of obviousness. MPEP 2143 says that to establish a prima facie case of obviousness, three basic criteria must *all* be met. First, there must be some suggestion or motivation, either in the reference themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. And, second, there must be a reasonable expectation of success. And, third, the prior art reference or references when combined must teach or suggest all the claim limitations. This third criterion has just been addressed above and all claims should be allowed on this basis alone. Nevertheless, consider the other two criteria for establishing a prima facie case of obviousness.

With regard to the rejection of claims 63-74 and to the first criterion in MPEP 2143 as applied to "admitted prior art", it is submitted that there is no suggestion or motivation within Abe and/or within so called "admitted prior art" to combine Abe with "admitted prior art". In the Office Action, for example on page 4, the Examiner admits that Abe does not explicitly disclose a schema and header files and Applicants agree. The Examiner then further suggests therein that "admitted prior art" discloses a schema in an analogous computer system and it would have been obvious to a person of ordinary skill in the art to combine these references because that person would be motivated to do so in order to include "prior art" header files within schema of an architecture to provide less arduous code generation, etc. Applicants respectfully disagree. The Examiner is merely pointing to the legacy software architecture challenge of the prior art, which was

presented by Applicants as background showing an unresolved industry problem requiring resolution, to support an argument for motivation for combining Applicants' disclosure with Abe. In Applicants view, this is an impermissible use of hindsight reconstruction from Applicant's disclosure. In addition, this suggested combination would also be improper because one of ordinary skill in the art would not have been motivated to combine the references for the following additional reasons.

The disclosure of Abe is limited to hardware architectures which operate to compile high level language into machine code and decompile machine code back into high level language. There is no need or place for a schema in Abe's computer environment. Abe's hardware architecture environment teaches away from the notion that a schema could be successfully or usefully integrated. Furthermore, Applicants' software architectures as expressed in Fig. 2A/B are within an object-oriented programming context. Abe does not disclose or suggest object oriented programming. To the contrary, the high level language in Abe is "C" which is NOT an object oriented language. *See* Abe, for example, column 4, line 43; column 6, lines 32, 37, 46 and other places where it refers to C; it also refers to FORTRAN and COBOL in column 7, lines 10-17, which also are not object oriented languages. Therefore, there simply is no motivation to be derived from within the *hardware-architecture-oriented, non-object-oriented-software Abe* to combine it with *software-architecture-oriented, object-oriented-software "admitted prior art"*, and vice versa. Rather, the two disclosures are not combinable, at least because their respective software is incompatible. Moreover, with respect to the second criteria in MPEP 2143, not only is there no reasonable expectation of success - the likelihood of success in attempting to mate incompatible software is zero.

Therefore, in view of the above, the combination of Abe and admitted prior art is improper, and the rejection of claims 63-74 which are rejected as being un-patentable over Abe in view of admitted prior art should be withdrawn and the claims allowed for these reasons in addition to those presented above with regard to Abe's not disclosing or suggesting software architecture.

Next, with regard to the rejection of claims 1-62 and 75-89 and to the first criterion in MPEP 2143 as applied to Bapat there is no suggestion or motivation within Abe and within "admitted prior art" to combine Abe and "admitted prior art" with Bapat or vice versa. In the Office Action, page 5, for example, it suggests that Bapat discloses certain of the elements of certain of Applicants' claims and cites certain passages in Bapat to that end. The Examiner continues:

> "Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of manipulating header files as taught by Bapat into the method of converting a program of a first architecture to a second architecture *as taught by the combination system of Abe and admitted prior art.* The modification would be obvious because of one of ordinary skill in the art would be manipulate the header files for mapping the within the schema of a database as suggested by Bapat (col 2, lines 6-17)." (Emphasis added.)

To begin with, Applicants have demonstrated above that Abe and "admitted prior art" are not combinable. Thus, there is no viable "combination system of Abe and admitted prior art", and that combination system with Bapat fails at the outset for that reason. Thus Bapat is irrelevant.

Moreover, the same object oriented software issue relative to Abe is found in Bapat, where Bapat could not be combined directly with Abe even if the "combination system" had somehow been viable (which it isn't). In other words, Bapat is not

compatible with Abe in any event because Bapat is based on object oriented programming (OOP); it specifically calls out the C++ language. See Bapat, for example, column 5, lines 51, 63, 68 and other places. But the high level language in Abe is "C" which is NOT an object oriented language as earlier noted. (*See* Abe, for example, column 4, line 43; column 6, lines 32, 37, 46 and other places where it refers to C; it also refers to FORTRAN and COBOL in column 7, lines 10-17, which also are not object oriented languages.) Although C++ is a superset of C, where use of a C++ compiler can make use of C language source code, the reverse is not true. C++ (the language in Bapat) cannot be directly converted into equivalent C language (the language in Abe). Thus, one of ordinary skill in the art reading Bapat would conclude that Bapat is irrelevant in this regard. Accordingly, there is no motivation to be derived from any of the references themselves to either combine Bapat with Abe directly or to combine Bapat with a non-existent "combination system". Obviously, if the references are un-combinable, then with regard to the second criterion in MPEP 2143, not only is there no reasonable expectation of success, there is absolutely no expectation of success.

In view of the above, it is submitted that the combination of Abe and "admitted prior art" and further in view of Bapat is improper. Therefore, the rejection of claims 1-62 and 75-89 should be withdrawn and the claims allowed for these reasons in addition to those presented above with regard to Abe's not disclosing or suggesting software architecture.

The determination of prima facie obviousness must be supported by a finding of "substantial evidence." See In re Zurko, 258 F.3d 1379, 1386 (Fed. Cir. 2001). Specifically, unless "substantial evidence" found in the record supports the factual

determinations central to the issue of patentability, including motivation and expectation of success, the rejection is improper and should be withdrawn. In view of the above, Applicants submit that that there is no "substantial evidence" in the record to support the factual determinations alleged by the Examiner with respect to motivation and/or expectation of success in combining "admitted prior art" with Abe, or in combining Bapat with both "admitted prior art" and Abe. Rather, the factual determinations alleged by the Examiner appear to have been arrived at by impermissible hindsight reasoning, as previously noted, after reading Applicants' disclosure and claims.

In response to the commentary in the Office Action, page 22, where the Examiner states that Applicants only make general allegations of improper hindsight reasoning and do not point out any errors in the rejection, it is submitted that the foregoing discussion amounts to substantially more than mere allegations of hindsight reasoning and does point out errors in the rejection.

In view of the above, Applicants submit that they have shown that a prima facie case of obviousness has not been made by the Examiner with regard to all claims (1-89).

Furthermore, in view of the above, Applicants submit that all independent claims (1, 12, 24, 37, 49, 52, 59, 63, 68, 70, 75 and 81) have also been shown to be allowable for reasons based on a separate argument. Thus, all dependent claims are likewise allowable at least by reason of their dependencies from their respective allowable independent claims.

Independent Claims 1, 12, 24, 37, 49, 52, 59 and 63
Management Software Issue

Reconsider the issue of management software. In the final Office Action, page 21, the "Examiner's response" to Applicants' prior amendment of August 2, 2004 states:

"Regarding management software, Abe system does provide a converting program for a computer of a first architecture to a machine program for a computer of second architecture whereas converting architecture of a program could be management software or program (col. 2, lines 32-45)." Applicants respectfully disagree that this could be reasonably interpreted as management software. The cited passage states:

> According to another aspect of the present invention, there is provided converting apparatus for converting an architecture of a program, comprising: a machine program producing means for *compiling a first high-level language* source program for a computer of a first architecture, *thereby producing a machine program for a computer of a second architecture*; a second high-level language source program producing means for *decompiling the machine program*, thereby producing a second high-level language source program which does not depend on any architecture; and a first executable load module producing means for *compiling and linking* the second high-level language source program, thereby producing a first executable load module. (Abe, col. 2, lines 32-45) (Emphasis added.)

This section of Abe merely discloses the following: (1) compiling a first high level language of a first hardware[3] architecture to generate machine language for a second hardware architecture, (2) decompiling that machine language to obtain a second high level language which does not depend on any hardware architecture, and (3) compiling and linking the second high level language. This is obviously not a discussion of the operation of management software. Management software is software which allows a customer to "monitor" and "change" the behavior of a piece of hardware. For example, management software monitors a storage system to see if all of its components are healthy and then changes the behavior of a component such as, e.g., increasing amount of

[3] Abe's disclosed architecture is hardware architecture as previously discussed; see, for example column 4, lines 32-59, where it discusses debugging a machine program ; it starts with an existing serial executing architecture and through compiling/decompiling etc. adapts this to a new parallel executing architecture. Clearly these are hardware architectures.

memory dedicated to write caching. By stark contrast, compiling, decompiling, and compiling/linking merely relate to machine code and higher level languages and do not have anything to do with, much less come close to, monitoring and changing the behavior of a piece of hardware. Therefore, the position taken in the final Office Action, as noted above, which equates the activity of compiling, decompiling etc. to management software is simply unreasonable and incorrect.

Furthermore, assuming, *arguendo*, that Abe's compiling/decompiling software *is* management software (a position with which Applicants clearly disagree) it *still* does not anticipate the management software recited in the claims. For example, claim 1 recites, *interalia*, "management software written in a first computer language compatible with first software architecture and not compatible with second software architecture", and Abe does not disclose or suggest any software with the software architecture compatibility/incompatibility as claimed, much less management software with that compatibility/incompatibility. Further, the management software of claim 1 utilizes header files: "header files being represented in said first language and capable of being utilized by said management software", and no such representation of header files being utilized by management software is made in Abe - Abe's compiler/decompiler does not make mention of header files. Bapat mentions header files, and Applicants' background mentions header files, but these "references" are not combinable with Abe for reasons previously provided herein.

Moreover, in addition to the compiling/decompiling irrelevancy to management software, there is no hint of management software by the software employed in Abe. Abe discloses a system employing "*debugging software*" compatible with first

architecture and (arguably) not compatible with second architecture. Debugging software is used BEFORE a system becomes operational; it is used to find and remove the bugs (problems) in a system that is under development or which has just been developed but which is not yet fully functional and without a history of successful operation. But, management software is used by a computer system which had previously passed through its debugging phase, and which is currently operational and functional. Management software can be used, for example, for keeping track of various functional operations involving storage, printers, servers, etc. within the computer system, as it is then running. The two categories of software, debugging software and management software are entirely different.

Applicants respectfully submit that the foregoing explanation has clearly demonstrated that Abe's compiler/decompiler technique is not management software. Furthermore, neither "admitted prior art" nor Bapat, each having been shown to be un-combinable with Abe, cures this management software deficiency of Abe. For the foregoing reasons, all independent claims reciting "management software" in both the preamble and the body of those claims, namely claims 1, 12, 24, 37, 49, 52, 59 and 63 are not disclosed or suggested by Abe taken alone or in combination with Bapat or admitted prior art. Accordingly, for these additional reasons, the 35 U.S.C. § 103(a) rejection of these independent claims, as well as their respective dependent claims, should be withdrawn and the claims allowed.

CONCLUSION

In view of the above arguments, the rejection of claims 1-89 under 35 U.S.C §103(a) should be withdrawn. To the extent that the above-discussed or any other Office

Action citations of Abe, "admitted prior art", and/or Bapat were applied against particular independent and dependent claim elements but not expressly rebutted herein, it is to be understood that Applicants' silence does not mean or imply acquiescence. Rather, Applicants believe that any such response to application of such citations would be moot in view of the foregoing arguments and provisions of MPEP § 2143.

Reconsideration and allowance of claims 1-89 are respectfully requested. This amendment should be entered under Rule 116 because it does not require the Examiner to do any further searching and it does not raise new issues for consideration. It further narrows-down the issues for appeal if the Examiner should not allow this application.

To the extent that an extension of time may be needed in order to enter this amendment in this case, please consider this response as including a petition under 37 C.F.R. § 1.136 for such extension of time. Please charge any fee for such petition or any other fee or cost that may be incurred by way of this amendment to Patent Office deposit account number **05-0889**. If the Examiner feels that a telephone conversation may serve to advance the prosecution of this application, he or she is invited to telephone Applicants' undersigned representative at the telephone number provided below.

Respectfully submitted,

JOEL WALL
Registration No. 25,648

TELEPHONE:
JOEL WALL ESQ.
508-625-1323
January 13, 2005